Aim :- To apply programming constructs of decision making and looping.

Objective :- To apply basic programming constructs like Branching and Looping for solving arithmetic problems like calculating factorial of a no entered by user at command prompt .

Theory :-

Programming constructs are basic building blocks that can be used to control computer programs. Most programs are built out of a fairly standard set of programming constructs. For example, to write a useful program, we need to be able to store values in variables, test these values against a condition, or loop through a set of instructions a certain number of times. Some of the basic program constructs include decision making and looping.

Decision Making in programming is similar to decision making in real life. In programming also we face some situations where we want a certain block of code to be executed when some condition is fulfilled. A programming language uses control statements to control the flow of execution of program based on certain conditions. These are used to cause the flow of execution to advance and branch based on changes to the state of a program.

- if
- if-else
- nested-if
- if-else-if
- switch-case
- break, continue

These statements allow you to control the flow of your program's execution based upon conditions known only during run time.

A loop is a programming structure that repeats a sequence of instructions until a specific condition is met. Programmers use loops to cycle through values, add sums of numbers, repeat functions, and many other things. ... Two of the most common types of loops are the while loop and the for loop. The different ways of looping in programming languages are

- while
- do-while

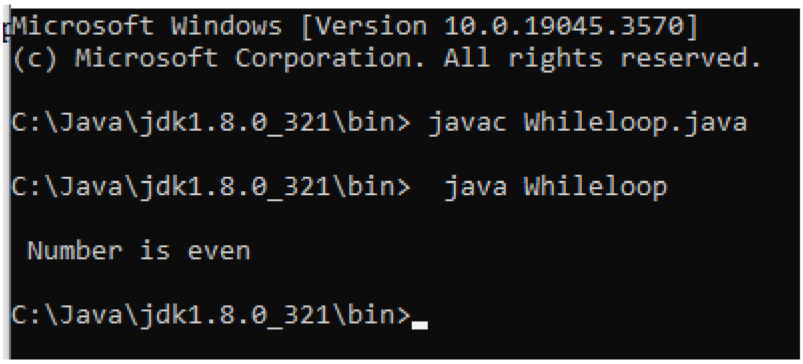
- for loop
- Some languages have modified for loops for more convenience eg :- Modified for loop in java.

For and while loop is entry-controlled loops. Do-while is an exit-controlled loop.

Code: -

1} while loop

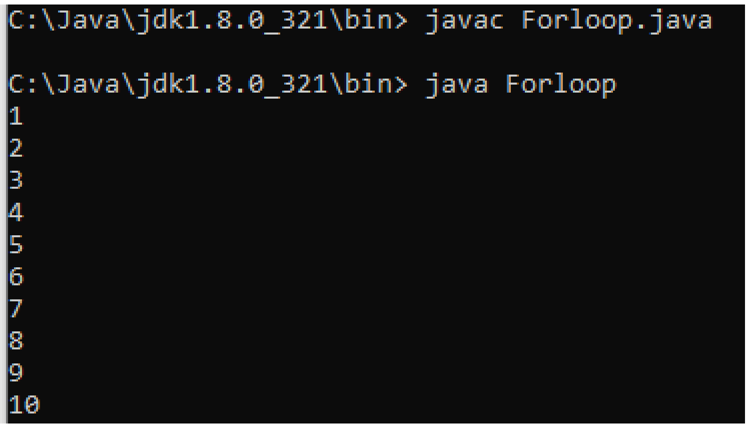
```
class Whileloop

{

  public static void main(String args[])

       {

         int a=4;

         while(a%2==0)

         {

           System.out.println("\n Number is even");

           break;

         }

       } }
```



```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Java\jdk1.8.0_321\bin> javac Whileloop.java

C:\Java\jdk1.8.0_321\bin>  java Whileloop

 Number is even

C:\Java\jdk1.8.0_321\bin>
```

2} for loop

```
class Forloop

{

  public static void main(String args[])

    {

      int x;

      for(x=1;x<=10;x++)

      {

        System.out.println(x);

      }

    }

}
```

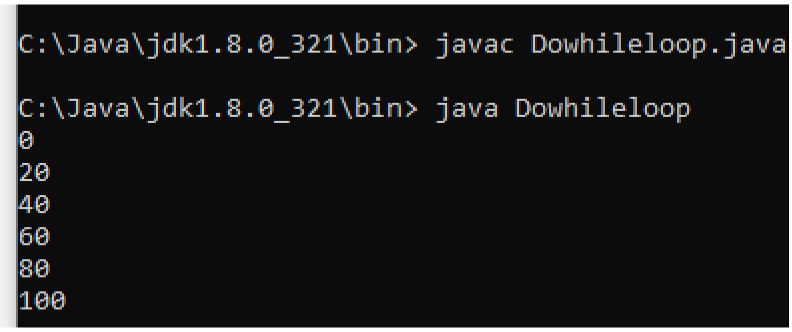


```
C:\Java\jdk1.8.0_321\bin> javac Forloop.java

C:\Java\jdk1.8.0_321\bin> java Forloop
1
2
3
4
5
6
7
8
9
10
```

3} dowhile loop

```
class Dowhileloop

{

   public static void main(String arg[])

  {

int a=0;
```

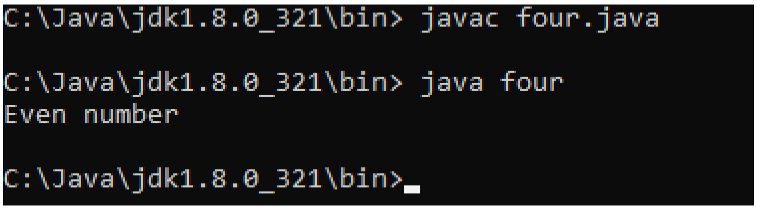
```
  do

 {

   if(a%20==0)

   {

    System.out.println(a);

    } a++;

 } while(a<=100);

 }

}
```



```
C:\Java\jdk1.8.0_321\bin> javac Dowhileloop.java

C:\Java\jdk1.8.0_321\bin> java Dowhileloop
0
20
40
60
80
100
```

4}if else

```
public class IfElseExample {

public static void main(String[] args) {

    int number=10;

        if(number%2==0){

        System.out.println("Even number");

    }else{

        System.out.println("Odd number");

    }

}
}
```



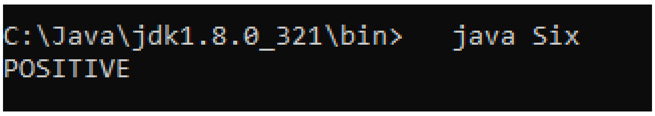
```
C:\Java\jdk1.8.0_321\bin> javac four.java

C:\Java\jdk1.8.0_321\bin> java four
Even number

C:\Java\jdk1.8.0_321\bin>
```

5} Ladder if else

```
class SecJavaProgram

{

 public static void main(String args[])

{

 int a=90;

if(a>=90)

{

System.out.println("grade A");

}

else if(a>=80)

{

System.out.println("grade B");

}

else if(a>=70)

{

System.out.println("grade c");

}

else if(a<70)

{

System.out.println("grade F");
}

}}
```

6} nested if else

```
public class PositiveNegativeExample {

public static void main(String[] args) {

    int number=15;

    if(number>0){

    System.out.println("POSITIVE");

    }else if(number<0){

    System.out.println("NEGATIVE");

    }else{

    System.out.println("ZERO");

   }

}}
```



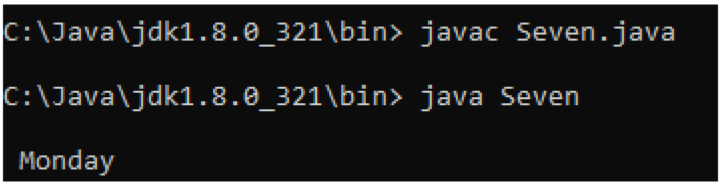
```
C:\Java\jdk1.8.0_321\bin>   java Six
POSITIVE
```

7} switch

```
class SwitchProgram

{

  public static void main(String args[])

         {

           int a = 1 ;

           switch(a)
```

```
{
case 1 :
   System.out.println("\n Monday");
   break;
case 2 :
   System.out.println("\n Tuesday");
   break;
case 3 :
   System.out.println("\n Wednesday");
   break;
case 4 :
   System.out.println("\n Thursday");
   break;
case 5 :
   System.out.println("\n Friday");
   break;
case 6 :
   System.out.println("\n Saturday");
   break;
case 7 :
   System.out.println("\n Sunday");
   break;
default :
   System.out.println("\n Not Valid");
}
```

```
} }
```



```
C:\Java\jdk1.8.0_321\bin> javac Seven.java

C:\Java\jdk1.8.0_321\bin> java Seven

 Monday
```

Conclusion:

1) Comment on how branching and looping useful in solving problems.

Branching and looping are fundamental control structures in Java (and many other programming languages) that are essential for solving a wide range of problems. They provide the means to make decisions and repeat actions, making your code more dynamic and adaptable.

Branching (if statements):

Decision Making: If statements allow you to make decisions in your code based on conditions. You can execute different blocks of code depending on whether a condition is true or false.

Looping:

Repetition: Loops (for, while, and do-while) enable you to repeat a block of code multiple times, which is useful for tasks like processing arrays, lists, and performing iterative calculations.